Pervasive and Mobile Computing 13 (2014) 85–98

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Adaptive content recommendation for mobile users: Ordering recommendations using a hierarchical context model with granularity☆

Jonghyun Han [a,1], Hedda R. Schmidtke [b,*], Xing Xie [c], Woontack Woo [d]

[a] Gwangju Institute of Science and Technology, 261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
[b] Carnegie Mellon University, Moffett Field, CA 94035, United States
[c] Microsoft Research Asia, No. 5, Danling Street, Beijing 100080, China
[d] KAIST UVR Lab., Daejeon, 305-701, Republic of Korea

ARTICLE INFO

Article history:
Received 11 January 2013
Received in revised form 1 November 2013
Accepted 11 November 2013
Available online 15 November 2013

Keywords:
Content recommendation
Search engine
Personalization
Context modeling
Mobile context-awareness
Context reasoning
Privacy-preserving context-aware systems

ABSTRACT

Retrieving timely and relevant information on-site is an important task for mobile users. A context-aware system can understand a user's information needs and thus select contents according to relevance. We propose a context-dependent search engine that represents user context in a knowledge-based context model, implemented in a hierarchical structure with granularity information. Search results are ordered based on semantic relevance computed as similarity between the current context and tags of search results. Compared against baseline algorithms, the proposed approach enhances precision by 22% and pooled recall by 17%. The use of size-based granularity to compute similarity makes the approach more robust against changes in the context model in comparison to graph-based methods, facilitating import of existing knowledge repositories and end-user defined vocabularies (folksonomies). The reasoning engine being light-weight, privacy protection is ensured, as all user information is processed locally on the user's phone without requiring communication with an external server.

1. Introduction

Mobile phones have become one of the most popular information browser devices, evolving into adaptive or personalized services for mobile users. Recent advances in light-weight reasoning techniques [1] promise to enable mobile phone users to interact with the available masses of data in a more efficient and satisfying way.

Mobile personalization systems have several shortcomings when compared to desktop-based systems, especially regarding computational power and screen size. As a consequence, it is difficult to process or display larger quantities of data on a mobile device. In contrast, they have a clear advantage regarding sensory equipment and usage: they are used in everyday life, not only in office-like environments. Thus, they have access to more of the user's personal information, which can be employed for personalization.

We propose an approach for efficiently retrieving and personalizing content relevant to the user's needs. We retrieve content related to a query, and then find user-preferred items from the retrieved content. We then show these results to

* Corresponding author. Tel.: +1 650 763 8018.
E-mail addresses: jhan@gist.ac.kr (J. Han), schmidtke@acm.org, schmidtke@cmu.edu (H.R. Schmidtke), xing.xie@microsoft.com (X. Xie), wwoo@kaist.ac.kr (W. Woo).
1 Tel.: +82 62 7153146; fax: +82 62 7152204.

http://dx.doi.org/10.1016/j.pmcj.2013.11.002
1574-1192/

the user for personalization. Our approach is to map mobile user context and properties of content onto a knowledge-based context model [1]. A modified reasoning engine allows us to compute semantic similarity between content and context using the context model. It uses internally a hierarchical data structure with size-based granularity. The context model stores three types of tags: tags that signify temporal and spatial contexts, such as '*Morning*' or '*Seoul*', and tags that purely indicate content, such as '*Baseball*' or '*Jazz*'. The context model also includes hierarchical information about these tags, e.g., that '*Seoul*' is in '*Korea*', and that '*Jazz*' is a type of '*Modern Music*'. During runtime, this data structure is used to represent mobile user context and properties of content, and to extract semantically related tags of measured context and content. We compute the priority of content based on similarity between measured context and content. The paper explores two variants of measuring similarity in the context model depending on whether size information is available or not.

We evaluate our approach with respect to a range of algorithms for retrieving content regarding the proposed context model. According to our evaluation, it is advantageous to use semantic context rather than raw context data. In general, the best algorithm utilized the proposed context model, which retrieves and uses semantically relevant terms of context. This method achieves a precision of 70% and a pooled recall of 46%. The difference in precision between our approach and the baseline algorithm is up to 22% points, and up to 17% points in pooled recall. Our approach performs fast enough to guarantee real-time processing on a mobile device and requires only slightly more time to configure (only in the initialization step, less than 25 ms) and explore (within 6 ms). Our evaluation indicates that the proposed approach is effective in advancing the reliability of personalized content recommendations on mobile devices.

The organization of the rest of this paper is as follows: Section 2 is devoted to analyzing related works. The hierarchical context model derived from [1] and its modified reasoning engine are presented in Section 3. In Section 4, the proposed approach for recommending mobile content based on user context is presented. The experiments and evaluations of the approach are shown in Section 5. Conclusion and future works are presented in Section 6.

2. Related works

In this section, we introduce several works related to the proposed recommendation approach. The features provided by the approach are based on various research areas, including context-awareness, content recommendation and personalization.

2.1. Context-awareness

Technologies for making systems context-aware can be used to personalize services or recommend user-centered content, as they enable systems to infer user interests and needs in a given situation. Xu Sun showed that spatial context (location information) can be used to improve mobile user experience by providing personally relevant mobile services [2]. Location-based services (LBS) are one type of context-aware service that takes advantage of location-sensing technologies and location information. In [3], a system that guides mobile users based on spatial, personal and social contextual information was presented. The system exploited sensors, such as GPS and history, obtained from a mobile phone.

Context-aware applications usually rely on a data structure or information repository called the *context model*, which handles the processing and abstraction of contextual information. Context models were designed to describe contextual situations and to represent semantic relations between context in order to allow applications to make use of this information. Since the first context models, numerous proposals have been put forth [4–15]. Over the years, context modeling approaches proceeded from the first tree-based models able to run on mobile phones and sensor nodes to approaches for querying fully-fledged ontologies stored on a remote server.

A key characteristic of context models is the inherent uncertainty that results if sensor measurements are the main sources of information. The inherent uncertainty of contextual information and how to address it has received considerable attention in the area of context modeling [6,14,16–19]. We adopted an interval-based approach [18,19] for modeling uncertainty in our system. Moreover, we employed partially defined levels of granularity, first proposed in [19], to allow representation of sizes in a manner that respects uncertainty of information obtained from users.

The context-aware application model underlying the approach used in this article, was first proposed by Jang et al. [9]. They suggest that the context of an interaction can be characterized by asking *who* interacts *when* and *where* with *what how* and *why*, the 5W1H questions [9]. In order to allow formal reasoning in 5W1H context models, the modeling language of [9] was later generalized and complemented with a formal semantics based on partial order reasoning, thus yielding a full-fledged decidable logical reasoning framework [20].

2.2. Reasoning about context

From the earliest location models [4], most context modeling approaches include not only a collection of contexts but also basic inferential capabilities. A context model can be a graph structure as in [4], then reasoning is implemented by following the edges of the graph. It can be a set of statements or filters as in [5]. Or it can be a full-fledged ontology with one or more reasoning mechanisms (cf. Bettini et al. [21] for an overview and in-depth discussion). Our approach is a combination of an ontology-based context model [22], founded upon the theory of mereotopology [23] widely used in ontology research, with a light-weight graph-based reasoning mechanism. Using our own light-weight graph-based mechanism has two advantages: first, all reasoning required for personalization, can be computed on the phone, so that potentially privacy-sensitive context data is not transmitted; second, using our own reasoning mechanism allows us to integrate the computation of context similarity, the key notion of our recommender system, directly into the reasoning mechanism.

Our context model does not directly re-use existing ontologies, but considerably overlaps with standard ontologies as it is founded upon basic concepts of mereotopology [22], a theory that underlies current standard ontologies and reasoning systems, in general [24]: mereotopology underlies, for instance, RCC [25], the Region-Connection-Calculus, and SUMO [26], the Suggested Upper Merged Ontology.

While we did not directly re-use existing ontologies for the prototype developed for this study, and a discussion of semantic information retrieval is clearly beyond the scope of this article, semantic information retrieval mechanisms can be a valuable source for importing information into our system. WordNet [27], SUMO [26] and other semantic information sources could be extracted or abstracted, perhaps automatically, with some seed concepts, so that a part could be used in our mobile application for recommendations. However, the approach presented in this paper is neutral with respect to the information source, the only strict requirement is the specific mereotopological basis upon which we build. This axiomatic basis for reasoning about context has been studied from a formal point of view [1,20,22,28] and has also been practically employed in a wide range of application areas, from monitoring industrial facilities [29] to preservation of business processes [30]. Any knowledge repository that is compatible with these concepts, including end-user defined repositories and folksonomies [31] could be applicable. In our prototype, the context model was developed by hand using public information repositories such as Wikipedia.[2]

2.3. Personalized content recommendation

Research into personalized content recommendation relates to the proposed approach as many recommendation systems use machine-learning technologies to select personalized content. In [32], a web recommender was introduced in which behaviors of users are modeled by constructing knowledge based on temporal web access patterns. It utilizes fuzzy logic to represent real-life temporal concepts and to construct a knowledge base of users' behaviors. A collaborative filtering method was used to improve the recommendation performance of electronic program guides in [33]. Yu-Cheng et al. proposed a community-based program recommendation [33]. This recommendation analyzed user habits and categorized users based on similar habits. These users were classified as belonging to specific communities in order to find and recommend programs.

Many resources can be used as information when recommendation systems model user behaviors for personalization. In [3,33–35], social information, such as community and friendship, were exploited to find user interests. Because social friends often have similar interests, social networks have become major resources for personalization. User histories of behavior, habits, and input also are used for inferring the next user interest. Shin et al. proposed a method that integrates user behavior profiling and content rating [36], while Bjelica [37] introduced a recommender system that efficiently learns the user's interests based on user modeling.

2.4. Information needs of mobile users

In order to provide preferred content to mobile consumers, we need to consider their information needs. Several studies have shed light on the types of information searched when consumers use a mobile search engine. [38–41] investigated the characteristics of mobile search by analyzing the log of search queries. Karen Church states that the personal nature of mobile devices is important for mobile searching [38]. Kamvar et al. [39,40] and Yi et al. [41] analyzed the patterns of search behaviors based on categorization of queries. In [42], both interests of users and mobile context, such as location and time, were found to be closely related to mobile users' information needs.

3. Hierarchical context model

Following the analysis of context modeling approaches and mobile user information needs, we concluded that a context model for personalized content recommendation for mobile users should contain knowledge about time, space, and content descriptions based on user interests. The context model is provided in a logical language format and represented internally in the form of directed acyclic graphs describing time, space, and general content or interests in terms of partial order hierarchies, implementing a mereological *part-of* relation [23] in different domains of context. Context tags are added as nodes in the graphs, and where available, coarse size information can be given with a tag.

During runtime, we map user context, such as sensor value and profile, onto the graph, and calculate the similarity between two context tags based on *specificity*. The key idea of the proposed approach is to find the most specific context tag of which both, the tags describing the user context and the tags describing a given content are part of; we then compute

[2] *Wikipedia*, http://www.wikipedia.org/.

Table 1
Domains of context.

Domain	Intended semantics	Relation
Time	Generalized time interval	Sub-interval: $\sqsubseteq_t$
Space	Region, scattered region	Sub-region: $\sqsubseteq_s$
Content	Classes of objects	Sub-class: $\sqsubseteq_c$

Table 2
Size and extension in different domains.

Domain	Extension	Relation
Time	Duration of an interval	Smaller duration: $\leq_t$
Space	Area of a region	Smaller area: $\leq_s$
Content	Number of objects in this class	Smaller cardinality: $\leq_c$

how close the user context and content context are, based on how specific this most specific common containing context is. The rationale behind this idea is simple: two users who share a very specific context are close, while users who share only an unspecific, generic context are more distant.

Take an example of spatial context: a user A is uploading a comment while standing in front of a picture in a museum in Paris in Fall 2009; in Summer 2011, another user B is standing in the same spot, looking for information. Our recommender system guesses that A's comment should be more relevant to B in this context than information contributed by a user C from a different location in the museum, where only the museum is the common spatial context: the area in front of the picture is a more specific common context.

We develop two methods for computing specificity and consequently similarity: a *graph-based method*, which uses the length of paths in the graph structure of the context model, and a *size-based method*, which employs size information given explicitly.

3.1. Structure

Information about the context of a mobile user can be obtained from sensors of a mobile phone, e.g., GPS yields latitude, longitude, and altitude coordinates; information about the context of content can be obtained from the tags attached to it. However, it is difficult to obtain meaningful information from the absolute values of a sensor signal, and therefore difficult to understand the semantic relation between a tag describing content and sensor data. For the example of geographic location: we need a location service to know that the GPS coordinate (37.566536, 126.977969) is in Seoul.

A knowledge-based context model can store such more abstract information. We implemented the context model using hierarchical graphs storing pre-order relations $\sqsubseteq_d$ between context tags that represent user context or properties of content. A pre-order relation $\sqsubseteq_d$ is a reflexive and transitive relation. Pre-order relations over a set T of tags can be represented in the form of directed acyclic graphs $G_d = (2^T, E_d)$ as a basis for efficient reasoning. The context model used in our implementation employs three partial order relations summarized in Table 1.

The nodes $v \in 2^T$ in the graphs are sets of contexts, and the relations $\sqsubseteq_d$ between contexts are represented by the edges $E_d \subset 2^T \times 2^T$. A path (v, w) exists in G_d iff there are $x, y \in T$ so that $x \in v$ and $y \in w$ and $x \sqsubseteq_d y$. The intuitive meaning is that tags in a set refer to the same entity with respect to the domain d, e.g.: '*Big Apple*', '*NewYork*', and '*New York*' are spatially indistinguishable.[3]

In addition to the relations $\sqsubseteq_d$ for the three domains, we allow coarse size information about the extension of a context in each domain to be represented. We interpret size information as duration for time, maximal diameter or area for space, and number of documents for content tags (Table 2). Information about extensions can be realized with a second directed acyclic graph structure (Fig. 1(a)). Edges in the first graph represent the relations in Table 1: the hour interval '*October 23rd, 2011 13:00*', for instance, is part of '*October 23rd, 2011 afternoon*', which in turn is part of '*October 23rd, 2011*'. Edges in the second graph represent the same-size-or-smaller-than relation ($\leq_d$), e.g., '*October 23rd, 2011*' has a size of '*24* h' or '*86,400* s' and has a shorter duration than '*Summer 2011*'.

Since an interval A that is part of another interval B cannot be larger than B, the graph representing the $\sqsubseteq_d$ relation is a subgraph of the graph representing size information. We therefore implemented both in a single data structure in our system.

More concretely, we write $G_d = (2^T, E_d)$ for a graph containing the $\sqsubseteq_d$-hierarchy, and $\Gamma_d = (2^{T \cup Y_d}, \Delta_d)$ for the corresponding graph containing the size hierarchy $\Delta \subset 2^{T \cup Y_d} \times 2^{T \cup Y_d}$, where Y_d is a domain specific set of numerical expressions over some given units, such as the hours and seconds in the example above for the domain of time. Reflecting the observation that a part cannot be larger than the entity it is part of, we demand that $E_d \subset \Delta_d$. Moreover, we assume

[3] It should be emphasized that being synonymous with respect to a one domain does not entail that the tags refer to the same entity with respect to other domains. With respect to content, for instance, '*Big Apple*' and '*NewYork*' might be distinguishable.

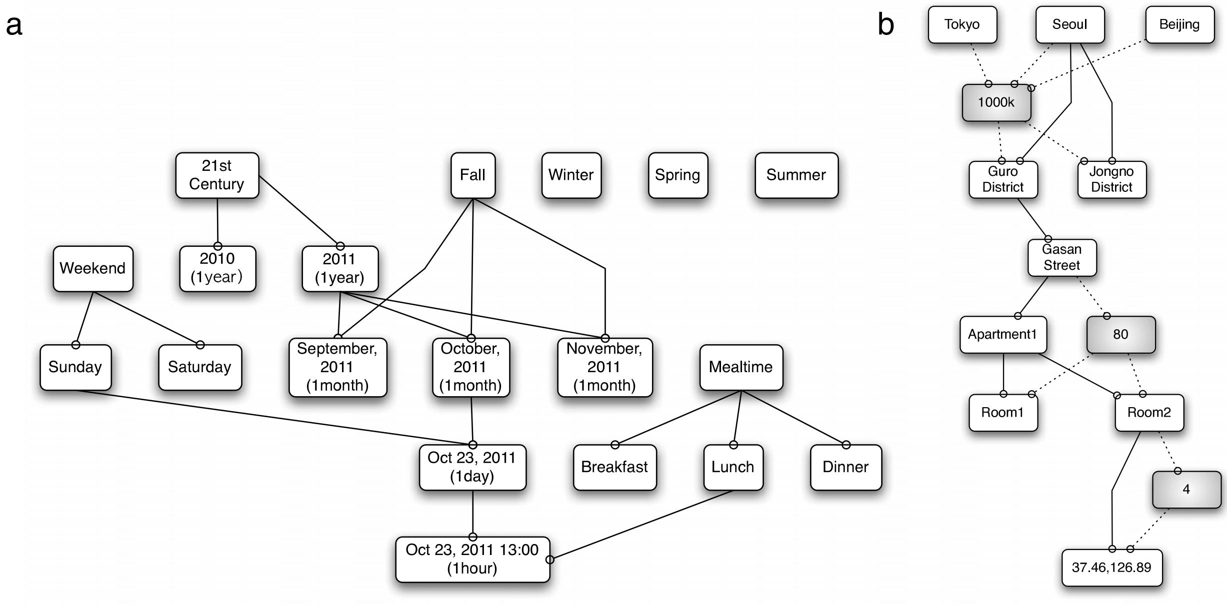


Fig. 1. Examples from a context model. *White* nodes represent context tags, edges in (a) represent a temporal part-of relation between context tags, while edges in (b) represent a spatial part-of relation and a smaller-than relationship. The dashed lines in (b) indicate edges that correspond to a smaller-than relation but not to a part-of relation, and *gray* boxes indicate nodes that represent size in the unit m^2.

that Δ_d also contains the same-size-or-smaller-than relation between numerical expressions $\upsilon \in Y_d$ over some given units, so that we can, for instance, compute that $(24 \text{ h}, 50 \text{ h}) \in \Delta_t$ from $24 < 50$.[4] We assume in the following that all size information is represented in a single unit, so that $\leq_d$ is a linear order with respect to all elements $\upsilon \in Y_d$ and that we can apply arithmetic as usual.

The graph representation allows us to store uncertainty about sizes. In Fig. 1(b), for instance, gray nodes represent spatial sizes $\upsilon \in Y_s$: the user in this example does not know the exact extension of the room '*Room1*', only that it has an area that is smaller than 80 m^2, but certainly larger than 4 m^2. For his apartment '*Apartment1*', he is not sure, whether it is smaller or larger than 80 m^2, but it can be inferred automatically that it is larger than '*Room1*', because '*Room1*' is a part of '*Apartment1*'.

It should be noted that directed acyclic graphs in contrast to trees allow nodes to have several parent nodes, that is a context can lie in the overlap of two contexts. For example, '*October 23rd, 2011 13:00*', obtained from a time-stamp, belongs to '*October 23rd, 2011*' and '*Lunch Time*' (Fig. 1). Transitive reasoning is implemented conveniently by following the edges in the graph. So, '*Weekend*' and '*Fall*' can be retrieved from a time-stamp '*October 23rd, 2011 13:00*' following the upper node '*October 23rd, 2011*'.

A second remark regards the semantics of spatially or temporally scattered entities: '*Baseball Field*', for instance, is spatially a scattered region consisting of the space occupied by all baseball fields; '*Lunch Time*' is a periodically reoccurring event temporally consisting of the intervals of all individual lunch times. We need entities like these to be represented in the context model, however, care must be taken to not confuse such scattered intervals/regions, which are the sum of separate convex intervals/regions, with sets of intervals/regions.

3.2. *Specificity of context*

In this hierarchical context model, more generic types of context intuitively have a tendency to be located in the upper levels, towards the root node of the graph, whereas more specific types of context are located in the lower levels further away from the root. To illustrate this intuition for the case of the three context domains:

- For time, an interval of long duration, such as the '*21st century*', would be on a higher level, the '*October of 2011*' would be on a lower level, and a short '*Shift-Key-Pressed*' event of 0.5 s duration would be at a very fine level.
- For the spatial domain, a spatial region with a large area, such as '*Europe*', would be on a high level, '*Rome*' on a lower level, and '*In Front of Notre Dame Cathedral*' on a fine level.
- In the case of content tags, generic types like '*Music*' or '*Photo*' are close to the root on a very high level with many documents, while '*Jazz of the Fifties*' would reside in a lower level with a smaller subset of music documents being tagged as such, and the '*Overture to Mozart's Magic Flute*' on a fine level.

4 More details on possible semantics for numerical computation and handling of units can be found in [22].

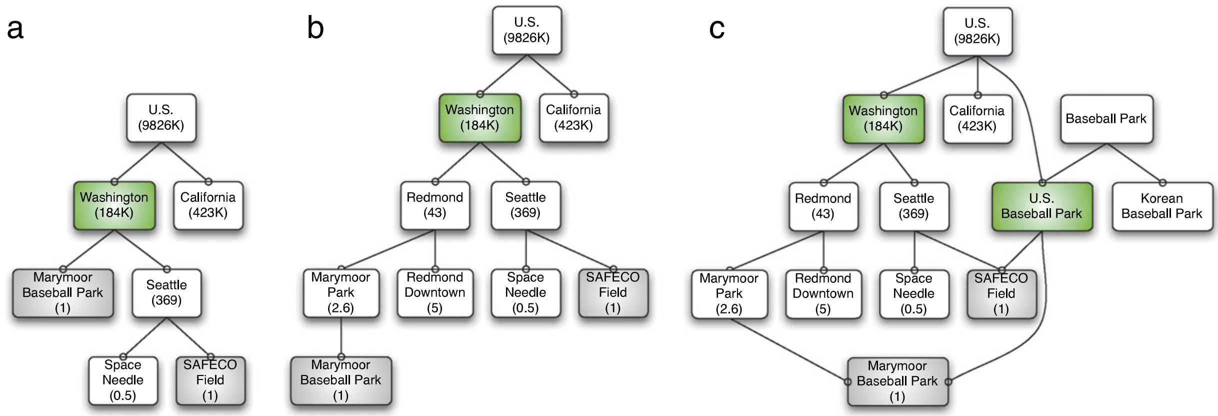


Fig. 2. Determining the least common subsumer (LCS) node (green) of two context nodes (gray) from specificity in three different context models with size information: (a) a simple tree-based context model, (b) a more detailed model that yields the same LCS node, and (c) different LCS nodes in a more complex model that is not a tree structure. Each context node is complemented with size information (in terms of the area, measured in km^2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

A range of approaches can be envisioned, in order to develop this intuitive idea of specificity into a formal notion. We explore two approaches: a graph-based approach where specificity is defined solely based on levels in the graph structure G_d, and a size-based approach where specificity is defined based on the size-constraints encoded in the graph Γ_d.

3.2.1. Graph-based method

In the graph-based method, the specificity $\mathrm{spec}_G(x)$ of a context x in a directed acyclic graph G_d is given as the depth of the node containing x normalized to the maximum depth of G_d:

$$\mathrm{spec}_{G_d}(x) = \frac{\mathrm{depth}(G_d, x)}{\mathrm{maxDepth}(G_d)}, \tag{1}$$

where $\mathrm{depth}(G_d, x)$ is assumed to be defined in the standard way as the length of the shortest path between x and the root node. As the maximum depth $\mathrm{maxDepth}(G)$ of the graph we take the length of the longest path between the root node and a context in the graph.

For instance, if we consider the graph in Fig. 2(a) to be our complete context model, then the specificity of '*Marymoor Baseball Park*' is $\frac{2}{3} = 0.67$, and the specificity of the node '*U.S.*' is $\frac{0}{3} = 0$, the most general node. However, the result of the graph-based method for calculating specificity highly depends on how detailed the view of the world in a particular part of the graph is modeled. For example, Fig. 2(a) and (b) show two context models describing similar spatial contexts under the node '*Washington*' at different levels of detail. However, the model in Fig. 2(b) is more detailed and has a greater depth. In this model, the specificity of '*Marymoor Baseball Park*' would be calculated as $\frac{4}{4} = 1$ and thus regarded as higher than in Fig. 2(a). Moreover, even the state '*Washington*' itself, which is roughly at the same position in both graphs receives different values for specificity – namely, $\frac{1}{3} = 0.33$ in (a) and $\frac{1}{4} = 0.25$ in (b) – simply because the second graph has greater depth.

3.2.2. Size-based method

If we want to allow for personalized and for incrementally changing user-defined context models, it should be less relevant, which of the models is more detailed. We therefore want to keep our assumptions about the quality and detail of context models as low as possible. A way to do this, which is also employed in tag clouds, is to weigh context tags using a measure of cardinality or size. From this idea we formulate the size-based notion of specificity.

If information about extensions is represented explicitly (Table 2), we receive a graph $\Gamma_d = (2^{T \cup Y_d}, \Delta_d)$ in which nodes can contain size information $\upsilon \in Y_d$. For $x \in T$ we define that a context $\upsilon \in Y_d$ is a d-extension of x, $\mathrm{ext}_d(x, \upsilon)$ if υ is larger or equally large as x in domain d and there is no smaller node $\upsilon \in Y_d$ that is larger or equally large as x.

$$\mathrm{ext}_d(x) = \min\{\upsilon \in Y_d \mid x \leq_d \upsilon\}. \tag{2}$$

Practically, we can compute $x \leq_d \upsilon$ as transitive inference, that is, by graph traversal from x along the edges in Δ_d.

Assuming all size information is given in a single unit, we can compute size-based specificity $\mathrm{spec}_{\Gamma_d}(x)$ of a context x based on the extension of x in d. We normalize the result again to be in the interval [0, 1], by comparing the extension with respect to the extension $\upsilon_{\max}$ of a fixed maximal context under consideration, e.g., the extension of the root node. The specificity of a context x is defined as 1 minus the ratio between the minimal extension of x and the size of the maximal region covered, as shown in Eq. (3).

$$\mathrm{spec}_{\Gamma_d}(x) = 1 - \frac{\mathrm{ext}_d(x)}{\upsilon_{\max}}. \tag{3}$$

In Fig. 2(a), the specificity of '*Washington*', the green node, is $1 - \frac{184{,}000}{9826{,}000} = 0.98$. If we use Fig. 2(b) the result is now the same as that in Fig. 2(a) and (c), irrespective of the intermediate level of detail represented in the model.

The main idea of our context model is that levels of sizes stratify the hierarchical context model in a way that is more neutral (ontology-independent), compared to the graph-based method. It depends less on an equally detailed view of the world and is thus closer to the way folksonomies are constructed, namely incrementally with different conceptual resolution in different parts. We use tags to distinguish and to retrieve information as necessary.

Consider the following example: a person in his hometown, an expert of the area, might have content tagged with concepts from a personal ontology of high detail with distinct, detailed descriptions for locations of his town and area; in contrast, an international traveler coming to the area, a layperson with respect to knowledge of the area, might use a range of descriptions at different levels of detail, loosely assembled into a collection of tags. Size information can help in this case, and it is easy to acquire, even if only in terms of rough comparisons as shown in Fig. 1.

3.3. Similarity between context nodes

We can now calculate the similarity of the current context of a mobile user and the context of a certain content by inspecting the most specific context nodes that describe them together, the *least common subsumers* (LCS) in a domain. A context z is an LCS of two contexts x and y in domain d, iff $x \sqsubseteq_d z$ and $y \sqsubseteq_d z$ hold and there is no other context z' for which this holds and that is between z and each of the two given contexts:

$$\mathrm{lcs}_d(x, y, z) \quad \text{iff} \quad x \sqsubseteq_d z \wedge y \sqsubseteq_d z \wedge \neg\exists z' : x \sqsubseteq_d z' \wedge y \sqsubseteq_d z' \wedge z' \sqsubseteq_d z. \tag{4}$$

It should be noted that there can be several nodes that qualify as LCS as illustrated in Fig. 2(c).

Graph-based and size-based similarity between two context nodes x and y can now be defined as the maximal specificity of LCS nodes of x and y:

$$\mathrm{sim}_{G_d}(x, y) = \max\{\mathrm{spec}_{G_d}(z) \mid z \in T, \mathrm{lcs}_d(x, y, z)\} \tag{5}$$

$$\mathrm{sim}_{\Gamma_d}(x, y) = \max\{\mathrm{spec}_{\Gamma_d}(z) \mid z \in T, \mathrm{lcs}_d(x, y, z)\}. \tag{6}$$

Because the specificity of a context is between 0 and 1, the similarity between two contexts is also between 0 and 1. The nearer two contexts are, the more similar they are. In other words, if the LCS node is very specific, the similarity is close to the maximum, 1. On the other hand, the more general the LCS nodes are, the lower the similarity is. The calculated similarity can be used to configure priorities of retrieved content.

Fig. 2 illustrates properties of the two variants for computing similarity. Consider, the node '*Marymoor Baseball Park*' is the most specific context describing the location of a mobile user p_1, and '*SAFECO Field*' is the most specific context node containing content c_1. Then, the LCS context that spatially encompasses both p_1 and c_1 is '*Washington*' in Fig. 2 (a) and (b). With the graph-based method, the similarity sim_{G_d} between p_1 and content c_1 would be rated $\frac{1}{3} = 0.33$ in Fig. 2(a), and $\frac{1}{4} = 0.25$ in Fig. 2(b). With information about the extension of the context, we can compute the similarity sim_{Γ_d}. The different level of detail in the modeling of the structures in the Fig. 2(a) and (b) does not affect the computed similarity between p_1 and c_1, which is $1 - \frac{184{,}000}{9826{,}000} = 0.98$ in both cases.

Fig. 2(c) shows an example model, in which the LCS is not unique. Here both '*U.S. Baseball Park*' and '*Washington*' are LCS nodes. For '*Washington*', the computed extension is again 184,000 km^2. The context '*U.S. Baseball Park*' is a spatially scattered region whose area is unknown in the model. The only information we can obtain in this case that it is smaller than the area of '*U.S.*' of which it is a part. The inferred spatial extension of '*U.S. Baseball Park*' therefore is 9,826,000, so that the most we can say is that spatially '*Washington*' is the only commonality, and similarity is again $1 - \frac{184{,}000}{9826{,}000} = 0.98$.

4. Personalized content retrieval

The proposed approach takes advantage of the similarity between mobile user context and properties of content to select user-preferred content. Both the mobile user context and properties of content are represented as context knowledge nodes in the hierarchical context model; the priorities of retrieved content are determined on the model based on the similarity of the nodes. To compute the similarity, we extract relevant terms of context and content property from the context model. Then, we compute the similarity of content based on the LCS node, which subsumes relevant nodes. Finally, we select user-preferred content according to their similarity.

The overall procedure of our approach is illustrated in Fig. 3. First, the context model proposed in Section 3 is configured on a mobile device. After a user inputs a search query, content is retrieved using the query and context is obtained from the mobile device's sensors. Then, a system using the proposed approach maps the content and context onto the context model in parallel. It retrieves relevant terms of the mapped content and context, respectively. It computes the similarity of the retrieved content and the obtained context, before selecting content based on the similarity. Finally, it displays the selected content on the mobile device.

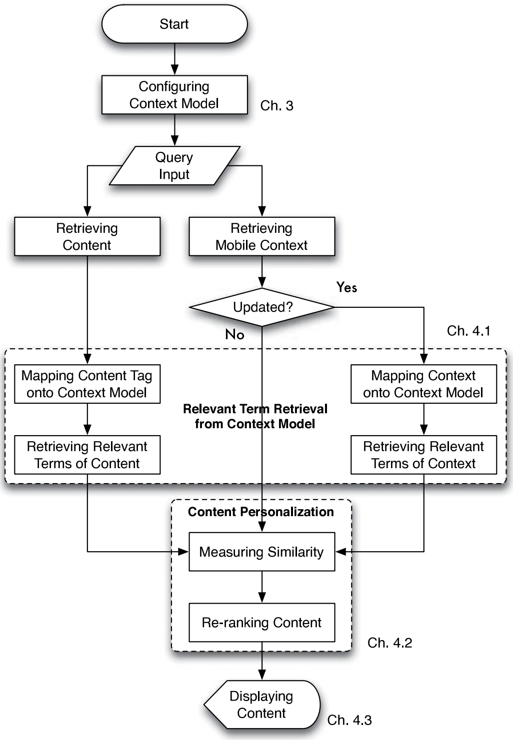


Fig. 3. The flow chart of the proposed approach.

4.1. *Retrieving relevant context knowledge terms*

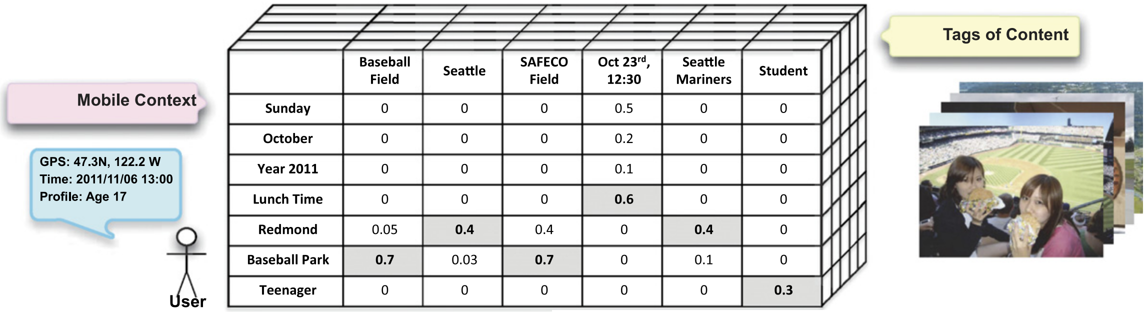
We map mobile user context, obtained from a mobile device, onto the hierarchical context model. The context belongs to one or more of the context nodes of the model based on pre-defined rules. For instance, rules, ('*Oct 23, 2011, 13:00*' $\sqsubseteq$ '*Oct 23, 2011*') and ('*13:00*' $\sqsubseteq$ '*Lunch time*'), create context '*Oct 23, 2011, 13:00*' from a time-stamp and link it into the context model, as demonstrated in Fig. 1. If the context is specific, it belongs to a context node located in a lower layer. More general context, in contrast, belongs to a context node located in an upper layer. We link measured context to the most specific node, so as to ensure specificity is computed correctly. The relation to ancestor nodes can be inferred.

We also map properties of content onto the context model. We assume the user is searching for content and provides only minimal information, such as 'History' or 'Comment'. Our system then retrieves all content that matches the query, from the database. Additionally, we retrieve information about each content, such as tags and spatial information. Then, they are mapped onto the context model as we did before with the mobile user context.

The information about content mapped onto the proposed context model is also used to extract relevant terms. From the hierarchical context model, we extract relevant context knowledge by exploiting the mapped nodes of context and properties of content. The relevant terms are ancestors of the nodes on the model. Because the meaning of a model's link is 'part-of', upper layer context knowledge involves lower layer context knowledge when they are connected. Upper layer context involves diverse content, but correlation among the content is low. In contrast, lower layer context knowledge involves a smaller range of content than upper context knowledge, but correlation among the content is high. For instance, photos taken in Europe are likely to include photographs of Paris as well as Rome. However, photos taken in Rome are more correlated than photos taken in Europe. Based on the extracted relevant terms located in the upper layer, we compute similarity of content and context.

4.2. *Personalizing content based on content similarity*

We determine the appropriateness of content for a mobile user by using extracted relevant contexts. The given mobile user context is used to determine appropriate content for the user in a given situation. For one-to-one correspondence between user context and content tags, we define set **U** as the set of context nodes related to mobile user context, and set **C** as the set of context nodes related to information about content. Then, we compute the content similarity between set **U** and set **C** through similarity on the context model. Eq. (7) describes how to compute similarity between context node set **U** and content node set **C**, where *ctx* is a member of set **U**, and *tag* is a member of set **C**. Fig. 4 shows an example of the computation. The left column of Fig. 4 represents relevant contexts retrieved from a mobile device, such as GPS, timestamp



	Baseball Field	Seattle	SAFECO Field	Oct 23rd, 12:30	Seattle Mariners	Student
Sunday	0	0	0	0.5	0	0
October	0	0	0	0.2	0	0
Year 2011	0	0	0	0.1	0	0
Lunch Time	0	0	0	**0.6**	0	0
Redmond	0.05	**0.4**	0.4	0	**0.4**	0
Baseball Park	**0.7**	0.03	**0.7**	0	0.1	0
Teenager	0	0	0	0	0	**0.3**

$$\mathrm{cntsim}(\mathbf{U},\mathbf{C}) = \frac{\sum_{tag\in\mathbf{C}} \max_{ctx\in\mathbf{U}}(\mathrm{sim}(tag,ctx))}{N(\mathbf{C})} = \frac{0.7+0.4+0.7+0.6+0.4+0.3}{6} = 0.52$$

Fig. 4. An example of similarity measurement between mobile context and content. The *left* column represents relevant terms of mobile context, and the *top* row represents relevant terms of content. Numbers are computed with one context term and one tag. *Gray* slots are selected maximum values.

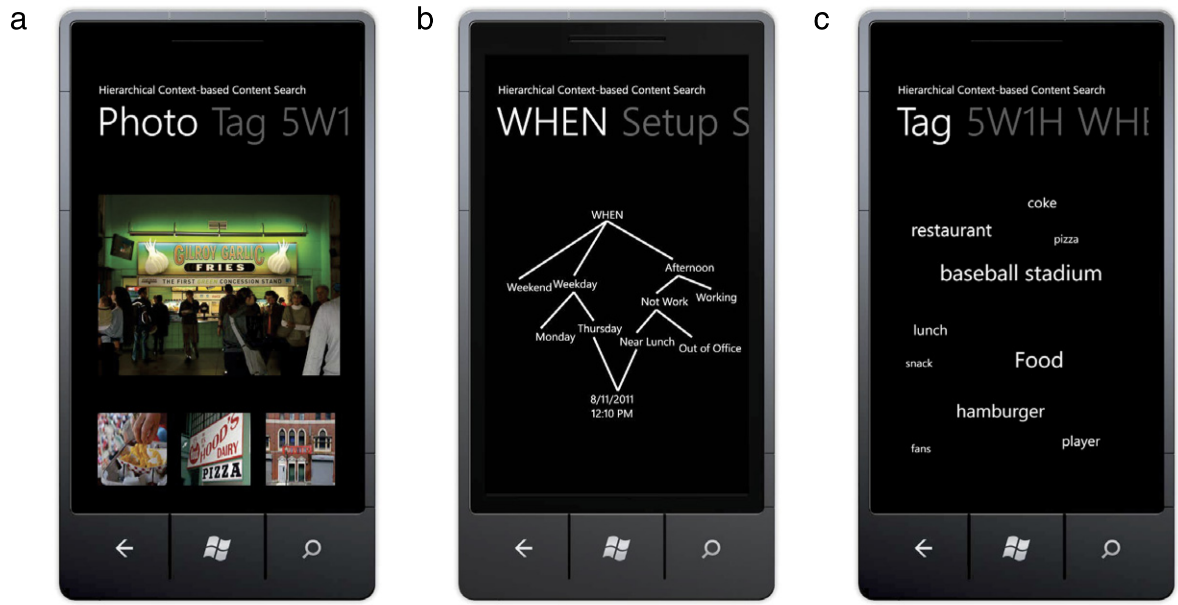


Fig. 5. User interface of photo recommender. (a) The list of retrieved photos. (b) Context model (c) Tag cloud of retrieved content.

and profile. The top row represents relevant content tags. We compute similarities between relevant contexts using Eq. (6), and select maximum values (Gray nodes in Fig. 4) of the similarities. Then, we compute the average of the values, 0.52 by Eq. (7). According to the proposed similarity method, we determine the priorities of retrieved content when providing a mobile user with the content.

$$\mathrm{cntsim}(\mathbf{U},\mathbf{C}) = \frac{\sum\limits_{tag\in\mathbf{C}} \max\limits_{ctx\in\mathbf{U}}(\mathrm{sim}_{\Gamma_d}(tag, ctx))}{|\mathbf{C}|}. \quad (7)$$

We re-rank the retrieved content according to the similarity between the content and user contexts. Our approach to re-ranking content is to compute similarity between nodes of content and nodes of context on a context model graph, and to re-rank the content according to the similarity scores. The similarity is estimated by Eq. (7) for each content and user context. After computing similarities of all retrieved contents, the contents are re-ranked in order of the similarity scores. Finally, we select the top five k to show to the user as content related to the user's preference and situation.

4.3. Displaying content on a mobile device

We implemented a system that uses our approach to recommend photo content related to an input query and user context. As shown in Fig. 5, we display selected photo content on a mobile device. The system initializes the context knowledge model first. Fig. 5(b) shows a part of the configured context model example. After a user inputs a query for search, the system retrieves photos from a web-based content sharing database. It applies the proposed personalization approach to the retrieved photos, and then displays selected photos. The thumbnails of recommended content are shown at the bottom of the screen, as shown in Fig. 5(a). A user is able to enlarge a photo from the recommended list. The rest of the recommended content is displayed if a user scrolls from side to side. The system provides additional information, such as photo tags and relevant contexts. Fig. 5(c) shows the tag cloud of recommended photos. It is possible to use a tag as a query for a new search. Through this system, we recommend personalized content with additional information.

5. Evaluation

In order to verify the suitability of the proposed approach, we evaluated the approach's results using the developed system. We checked the availability of the context model on a mobile device and compared the appropriateness of personalized content results with other approaches.

5.1. Settings

Mobile Device. In this evaluation, we used a smartphone equipped with a GPS sensor as a mobile device. It contained a 1 GHz single-core CPU. Users could input a user profile on the mobile device before starting our system. The user profile was compiled with user context and sensor data.

Data Set. We used photos as content to provide to a mobile user. The photos were stored on a web-based photo sharing service, *Flickr*.[5] To exploit spatial information of content, we only used geo-tagged photos containing latitude and longitude. The additional information about the photos, such as tags and timestamp, also were used. The number of retrieved photos per query was 50, and a photo had 4.6 tags, on average. We took advantage of geocoding to convert latitude and longitude values to names of regions, and used the names as tags.

To evaluate the results of the proposed approach, we distributed our system to 10 participants for a week. We recruited university students, staff and graduate students. We selected participants who did not have knowledge related to this work. Among the participants, 8 participants are male and 2 participants are female. Their ages range from 22 to 36, and the average of their age is 29.3. They were experienced in image search using mobile devices. The number of queries used by the participants totaled 74. In order to make certain that situations were natural, we did not set a limit on the usage of the system. Instead, we recorded logs of sensor values of a GPS and a timestamp; participants freely recorded things that they wanted to find in this diary.

We generated a context knowledge model manually. As shown in Fig. 1, we generated context nodes that had a hierarchical structure. In order to get context tags and their links, we referred to the *Open Directory Project*[6] (ODP). We also referred to *Wikipedia*[7] information for getting knowledge related to granularity of the context tags. We restricted temporal context nodes to Oct, 2011, and restricted spatial context nodes to South Korea, because our evaluation was performed in South Korea in autumn 2011. The generated nodes captured the real context for describing situations when participants used our system. When we evaluated the impact of the number of context knowledge nodes, we randomly selected context nodes from the manually generated nodes. We iterated 10 times when the randomly generated models were used.

5.2. Approach of evaluation

In order to provide the first evaluation of the feasibility of the proposed context model, we tested the model's performance on a mobile phone. We compared the elapsed time to explore the context model to find relevant contexts in models having different numbers of context nodes. We measured the amount of time it took to retrieve relevant contexts from a context model. This task was necessary to compute the similarity between content and context.

We compared our approach (algorithm PROPOSED) with the following five algorithms: As a baseline algorithm (A1-BASE), instead of using mobile user context, the number of matched tags with queries was used to decide the ranking of content. Another algorithm (A5-CONT) used all context: spatial, temporal and personal information without the proposed model. The others (A2-SPAT, A3-TEMP, A4-PERS) used each context separately without the model. They used only absolute values of context, not the proposed model.

To evaluate different algorithms on the same data set, participants evaluated all of the retrieved content off-line after using our system. They were supplied with queries, the lists of content (in a random order) and sensor values recorded when the participants used our system. Sensor values consist of a timestamp (dd-mm-yy hh: mm), a GPS coordinate (latitude, longitude), and orientation by a compass (angle). In order to help participants to understand the values, we also provided a map marked the point of the GPS and the compass, together with the name of places by reverse geocoding. We did not give any information related to algorithms that recommend the photos. The lists of content contained photos and tags with a

5 *Flickr*, http://www.flickr.com/.

6 *Open Directory Project*, http://www.dmoz.org/.

7 *Wikipedia*, http://www.wikipedia.org/.

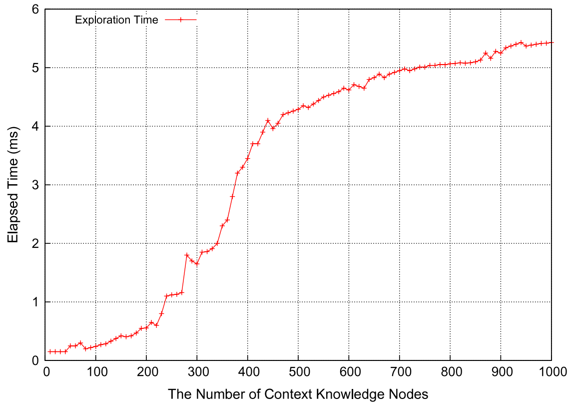


Fig. 6. Evaluation result showing the elapsed time when the context model is explored to find relevant contexts per query.

score box. Thus, the participants simply input a score for a photo into the box. They used the following scoring instructions to decide when a photo was relevant to their interests in a given situation:

– *You entered the query 'x' to our system previously in (day-month-year time) at (the name of places). To remember the situation that you entered the query, please refer to given sensor values. In that time, these photos could be obtained and provided from a database. Based on the following instructions, please score the appropriateness of each photo when you were provided with the photo in the given situation. Please consider your interest in the given situation when you entered the query.*

- 1, *if the photo is not related to the query;*
- 2, *if the photo is about the query in general, but not your interest;*
- 3, *if the photo is about the query in general, and related to your interest in general;*
- 4, *if the photo is about the query, and related to your interest in general;*
- 5, *if the photo is about the query, and related to your interest exactly.*

If a photo was given a score of 3 or higher by a participant, we regarded the photo as "relevant" (A). We regarded the photo as "very relevant" (A+) if its score was 4 or 5.

We used precision and recall to evaluate our approach and the comparison algorithms. Precision was the ratio of returned relevant photos to returned photos as shown in Eq. (8). As Eq. (9), recall was the ratio of returned relevant photos to relevant photos. When computing precision and recall, we used pools instead of all retrieved photos in our calculations [43]:

$$precision = \frac{|R \cap P|}{|P|} \tag{8}$$

$$recall = \frac{|R \cap P|}{|R|} \tag{9}$$

where R is the set of all relevant photos and P is the set of retrieved photos.

In addition, we also employed the normalized discounted cumulative gain (nDCG) to evaluate the effectiveness of our approach [44]. It is often used in information retrieval according to relatively ideal positions. In order to measure the appropriateness of re-ranked content, we computed nDCG values based on evaluated scores by participants. We considered the ranks given by participants as ideal results for normalization.

5.3. Results

We first evaluated the performance of our proposed context model for availability on a mobile device. We measured time performance when a system explored a context model to find relevant contexts. As an online processing task, it needs to guarantee real-time processing speed. Fig. 6 shows the results of time performance. The x-axis is the number of context nodes in the model. As shown in Fig. 6, there was a sharp increase in the elapsed time to explore context nodes when the number of context nodes was between 300 and 500. However, if the number of context nodes was larger than 500, there was a slight increase, compared to fewer context nodes. The elapsed time was less than 6 ms even though the number of contexts exceeded 1000. Because the structure of our proposed model is similar to a tree structure, the exploration time increased only slightly when the number of nodes was larger. Therefore, performance was fast enough to guarantee real-time retrieval processing on the phone.

In order to verify the reliability of our recommended results, we compared photos retrieved by the proposed approach with the results of the algorithms described in Section 5.2. Table 3 presents the precision and pooled recall for all the different photo retrieval algorithms, i.e., percentage of photos exactly on a query. All algorithms exploiting context performed better than the baseline algorithm (A1), which did not use any context. The algorithm that had the highest precision and recall

Table 3
The evaluation results of precision and recall.

Algorithm	A		A+	
	Precision	Recall	Precision	Recall
A1-BASE	48%	29%	31%	21%
A2-SPAT	59%	38%	39%	29%
A3-TEMP	56%	37%	36%	25%
A4-PERS	60%	37%	40%	30%
A5-CONT	61%	38%	41%	31%
PROPOSED	70%	46%	49%	35%

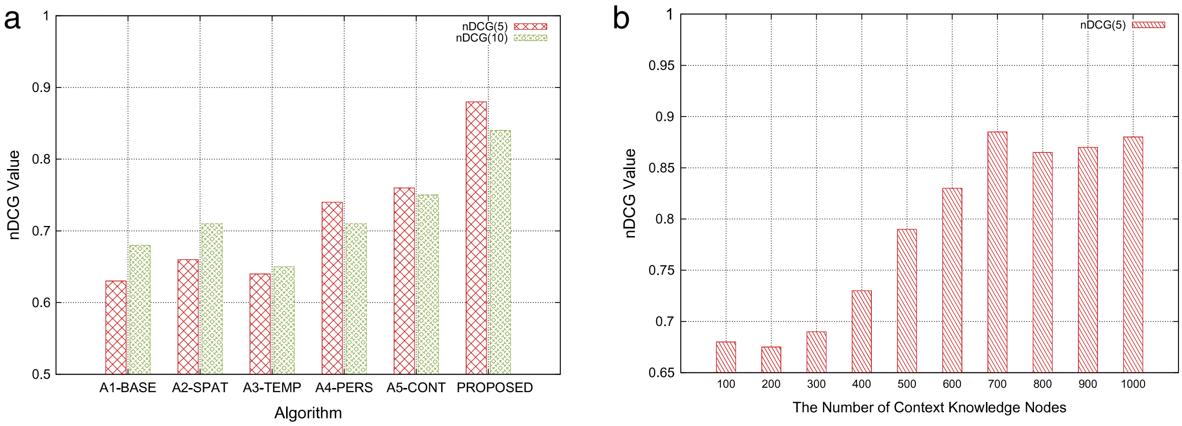


Fig. 7. Evaluation results showing the nDCG value. (a) Comparison of the nDCG by different algorithms when retrieving 5 items and 10 items. (b) Comparison of the nDCG@5 by the number of context knowledge nodes.

value was the proposed approach. When we used a set of relevant photos (**R**), i.e., the percentage of photos with score of 3 and over, the precision of our approach reached 70%, and the recall was 46% compared to 29% from the baseline algorithm. When we used a set of very relevant photos (**R+**), the precision and recall of our approach was higher than the other algorithms compared in this evaluation. Therefore, the table indicates that context improves content recommendations and the proposed context model is effective in enhancing context usage.

Fig. 7(a) compares the five algorithms and the proposed approach based on nDCG. Apart from the proposed approach, the most effective algorithm was A5, exploiting all context without the hierarchical context model. The nDCG values of our approach exceeded 0.8, whether we used the first 5 retrieved photos or 10 photos. They were higher than A5 by 0.12 and 0.09 respectively. This finding indicates that content re-ranking of our approach enhances the order of retrieved content because the nDCG value is affected by the order of retrieved items. Although the value of nDCG@10 was lower than nDCG@5 in the case of our approach, it was higher than the other algorithms. The nDCG@10 of our approach was lower than nDCG@5 because the additionally retrieved 5 photos by nDCG@10 were more generalized than the commonly retrieved 5 photos. Through Fig. 7(a), we can ascertain that our approach is effective for the order of content, and that fewer retrieved items are more appropriate in our approach.

In addition, we evaluated nDCG@5 values with different numbers of context nodes. The bar graph of Fig. 7(b) shows nDCG values of our approach with 5 retrieved photos. Overall, if the number of context nodes was increased, nDCG values also were increased. When we used 700 context nodes for our approach, nDCG value was maximized. Adding more nodes did not improve results further. Rather, the nDCG remains quite stable without significantly decreasing when there were more than 700 context nodes, although context nodes added in the upper layer lead to more general content retrieved. From Fig. 7, if we use more than 500 context nodes with our approach, the retrieved content is more appropriate than the other algorithms.

5.4. Discussion

Overall, context helps recommender systems understand the information needs of mobile users. Compared to the baseline algorithm, the precision, recall and nDCG were improved when context was used (A2, A3, A4, A5, and our approach). Among algorithms that use context, our approach is the most effective, although it requires slightly more time to configure and explore a context model, as shown in the evaluation. Because our hierarchical context model is able to provide contexts that are semantically similar with a query and context, it enhances measuring similarity between content and context considerably in comparison to the other algorithms.

Here, we give examples to illustrate the effectiveness of our approach. For instance, when a user searches '*food*' in '*SAFECO Field*' baseball stadium at noon, the baseline algorithm provides photos including the '*food*' tag in the order of popularity. The other algorithms exploiting context without our model filter photos taken far from the stadium or photos taken at different times of day. The photos that satisfy context conditions are more appropriate for the user; however, the problem is that there are too few of these photos. Our approach recommends not only photos taken in '*SAFECO Field*', but also photos dealing with fast food in other American baseball stadiums at lunch time according to personal preference (the user frequently uploaded photos tagged '*fast food*' before). Fig. 5 shows the example of retrieved photos and the relevant context tags used by our approach. The number of retrieved photos via our approach is sufficient compared to the other approaches. Our approach is able to provide content that is more appropriate for a given situation and more familiar to users. Table 3 and Fig. 7 show the appropriateness of recommended photos using our approach.

However, there are a number of limitations to our approach. Although we can measure consistent similarity between context nodes regardless of the level of detail in a context model using size-based granularity, a highly-detailed model is better at extracting relevant contexts. As shown in Fig. 7(b), the number of context nodes affects the relevance of retrieved content because more relevant contexts retrieved by the model make measuring similarity between context and content more accurate. It follows that users highly benefit from detailed contextual knowledge. This could be an incentive for users to create and share their expert knowledge with others. Tools for easy maintenance and sharing are needed to realize this.

A positive feature of our approach is that adding more and more nodes, the system reaches a relatively stable nDCG value of around 0.87 at around 700 nodes. In contrast to graph-based approaches, the size-based method for computing similarity between nodes, keeps the behavior of the system stable once the maximum is reached even as highly generic nodes or intermediate nodes are added randomly to the graph.

To verify the actual usefulness of the developed mobile system, it would be necessary to conduct a qualitative analysis, such as a usability study for the system usage. We concentrated on the relevance of retrieved photo content in our evaluation. This evaluation is helpful in understanding the accuracy of the recommended content, but we need to evaluate the ease of use of the developed system. Our current approach can be improved by considering these issues.

6. Conclusions and future work

In this paper, we introduced an approach for recommending personalized content to mobile users through a hierarchical context model with size-based granularity information. The proposed context model is organized as a hierarchical directed acyclic graph with size information. We use it to compute similarity between content and context for personalization. Our research goal was to retrieve content effectively in order to help mobile users browse information. The evaluation shows that the proposed approach is effective in personalizing and recommending relevant content through exploiting contextual information. We expect that our approach can improve information browsing by personalizing content effectively on a mobile device.

The context model for our current prototype was developed by hand from publicly available information sources, such as Wikipedia. In ongoing and future work, we explore the possibilities for information integration and automatic information retrieval, e.g. from standard ontologies and other knowledge repositories, such as SUMO [26] or WordNet [27], but also from folksonomies [31] and other knowledge repositories created by end-users. Using size-based granularity to compute similarity instead of a structural, graph-based method, our approach can work with ontologies or folksonomies of varying conceptual density.

In contrast to other recommender systems, which require user information to be sent and stored on a server, privacy protection is a key feature of our approach. Context libraries and contents can be shared as a user wishes, but personal information from sensors giving the current context of the user always stays on the user's phone, since the reasoning engine is light-weight, designed to work locally.

Acknowledgments

This work is supported by the IT/SW Creative Research Program 2011, funded by Ministry of Knowledge Economy and Microsoft Research Asia. We are very grateful for the valuable input provided by our reviewers on earlier versions of this article.

References

[1] H.R. Schmidtke, Contextual reasoning in context-aware systems, in: Workshop Proceedings of the 8th International Conference on Intelligent Environments, IOS Press, 2012, pp. 82–93.
[2] X. Sun, A. May, The role of spatial contextual factors in mobile personalization at large sports events, Personal Ubiquitous Comput. 13 (2009) 293–302. http://dx.doi.org/10.1007/s00779-008-0203-6.
[3] Y. Suh, C. Shin, W. Woo, A mobile phone guide: spatial, personal, and social experience for cultural heritage, IEEE Trans. Consum. Electron. 55 (4) (2009) 2356–2364. http://dx.doi.org/10.1109/TCE.2009.5373810.
[4] B.N. Schilit, N.I. Adams, R. Want, Context-aware computing applications, in: Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, 1994, pp. 85–90.
[5] D. Salber, A.K. Dey, G.D. Abowd, The context toolkit: aiding the development of context-enabled applications, in: CHI, ACM Press, Pittsburgh, 1999, pp. 434–441.

[6] A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K.V. Laerhoven, W.V. de Velde, Advanced interaction in context, in: Handheld and Ubiquitous Computing, 1999, pp. 89–101.
[7] A.K. Dey, D. Salber, G.D. Abowd, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Hum. Comput. Interact. J. 16 (2–4) (2001) 97–166.
[8] A. Ranganathan, R. Campbell, An infrastructure for context-awareness based on first-order logic, Personal Ubiquitous Comput. 7 (6) (2003) 353–364.
[9] S. Jang, W. Woo, Ubi-UCAM: a unified context-aware application model, in: Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT'03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 178–189.
[10] G. Chen, Solar: Building a context fusion network for pervasive computing, Ph.D. Thesis, Dartmouth College, Hanover, NH, USA, 2004.
[11] J.E. Bardram, The Java context awareness framework (JCAF)—a service infrastructure and programming framework for context-aware applications, in: H.-W. Gellersen, R. Want, A. Schmidt (Eds.), Pervasive Computing, 2005, pp. 98–115.
[12] T. Gu, H.K. Pung, D.Q. Zhang, A service-oriented middleware for building context-aware services, J. Netw. Comput. Appl. 28 (1) (2005) 1–18.
[13] S.W. Loke, Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective, Knowl. Eng. Rev. 19 (3) (2005) 213–233.
[14] K. Henricksen, J. Indulska, Developing context-aware pervasive computing applications: models and approach, Pervasive Mobile Comput. 2 (2006) 37–64.
[15] J. Ye, L. Coyle, S. Dobson, P. Nixon, Representing and manipulating situation hierarchies using situation lattices, Rev. d'Intelligence Artificielle 22 (5) (2008) 647–667.
[16] P.D. Gray, D. Salber, Modelling and using sensed context information in the design of interactive applications, in: M.R. Little, L. Nigay (Eds.), Engineering for Human-Computer Interaction, 2001, pp. 317–336.
[17] D. Chalmers, N. Dulay, M. Sloman, Towards reasoning about context in the presence of uncertainty, in: Workshop on Advanced Context Modelling, Reasoning And Management, Nottingham, UK, 2004.
[18] A. Ranganathan, J. Al-Muhtadi, R.H. Campbell, Reasoning about uncertain contexts in pervasive computing environments, IEEE Pervasive Comput. 3 (2) (2004) 62–70.
[19] H.R. Schmidtke, W. Woo, A formal characterization of vagueness and granularity for context-aware mobile and ubiquitous computing, in: H.Y. Youn, M. Kim, H. Morikawa (Eds.), International Symposium on Ubiquitous Computing Systems, Springer, Heidelberg, 2006, pp. 144–157.
[20] H.R. Schmidtke, D. Hong, W. Woo, Reasoning about models of context—a context-oriented logical language for knowledge-based context-aware applications, Rev. d'Intelligence Artificielle 22 (5) (2008) 589–608.
[21] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive Mobile Comput. 6 (2) (2010) 161–180.
[22] H.R. Schmidtke, W. Woo, Towards ontology-based formal verification methods for context aware systems, in: H. Tokuda, M. Beigl, A. Brush, A. Friday, Y. Tobe (Eds.), Pervasive 2009, Springer, 2009, pp. 309–326.
[23] B. Smith, Mereotopology: a theory of parts and boundaries, Data Knowledge Eng. 20 (1996) 287–303.
[24] J. Bateman, S. Farrar, Spatial ontology baseline, Tech. Rep., Collaborative Research Center for Spatial Cognition. I1-[OntoSpace] D1, 2004.
[25] D. Randell, Z. Cui, A. Cohn, A spatial logic based on region and connection, in: Knowledge Representation and Reasoning, Morgan Kaufmann, 1992, pp. 165–176.
[26] I. Niles, A. Pease, Towards a Standard Upper Ontology, in: FOIS, 2001, pp. 2–9.
[27] G.A. Miller, Wordnet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[28] H.R. Schmidtke, M. Beigl, Distributed spatial reasoning for wireless sensor networks, in: Modeling and Using Context, Springer, 2011, pp. 264–277.
[29] Y. Ding, F. Laue, H.R. Schmidtke, M. Beigl, Sensing spaces: light-weight monitoring of industrial facilities, in: Workshop Proceedings of the 5th International Workshop on Behaviour Monitoring and Interpretation, vol. 678, CEUR, 2011, pp. 53–62.
[30] M.A. Neumann, T. Riedel, P. Taylor, H.R. Schmidtke, M. Beigl, Monitoring for digital preservation of processes, in: Modeling and Using Context, Springer, 2011, pp. 214–220.
[31] A. Mathes, Folksonomies—Cooperative Classification and Communication Through Shared Metadata, Tech. Rep., Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, 2004.
[32] A. Fong, B. Zhou, S. Hui, G. Hong, T.A. Do, Web content recommender system based on consumer behavior modeling, IEEE Trans. Consum. Electron. 57 (2) (2011) 962–969. http://dx.doi.org/10.1109/TCE.2011.5955246.
[33] Y.-C. Chen, H.-C. Huang, Y.-M. Huang, Community-based program recommendation for the next generation electronic program guide, IEEE Trans. Consum. Electron. 55 (2) (2009) 707–712. http://dx.doi.org/10.1109/TCE.2009.5174443.
[34] J. Han, X. Xie, W. Woo, Context-based microblog browsing for mobile users, J. Ambient Intelligence Smart Environ. 5 (1) (2013) 89–104.
[35] J. Han, X. Xie, W. Woo, Context-based local hot topic detection for mobile user, in: Adjunct Proceedings of the 8th International Conference on Pervasive Computing, Springer, 2010, pp. 5–8.
[36] H. Shin, M. Lee, E. Kim, Personalized digital TV content recommendation with integration of user behavior profiling and multimodal content rating, Consumer Electronics, IEEE Transactions on 55 (3) (2009) 1417–1423. http://dx.doi.org/10.1109/TCE.2009.5278008.
[37] M. Bjelica, Towards TV recommender system: experiments with user modeling, IEEE Trans. Consum. Electron. 56 (3) (2010) 1763–1769. http://dx.doi.org/10.1109/TCE.2010.5606323.
[38] K. Church, B. Smyth, K. Bradley, P. Cotter, A large scale study of European mobile search behaviour, in: Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI'08, ACM, New York, NY, USA, 2008, pp. 13–22. http://doi.acm.org/10.1145/1409240.1409243.
[39] M. Kamvar, S. Baluja, A large scale study of wireless search behavior: google mobile search, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'06, ACM, New York, NY, USA, 2006, pp. 701–709. http://doi.acm.org/10.1145/1124772.1124877.
[40] M. Kamvar, M. Kellar, R. Patel, Y. Xu, Computers and iPhones and mobile phones, oh my!: a logs-based comparison of search users on different devices, in: Proceedings of the 18th International Conference on World Wide Web, WWW'09, ACM, New York, NY, USA, 2009, pp. 801–810. http://doi.acm.org/10.1145/1526709.1526817.
[41] J. Yi, F. Maghoul, J. Pedersen, Deciphering mobile search patterns: a study of Yahoo! mobile search queries, in: Proceeding of the 17th International Conference on World Wide Web, WWW '08, ACM, New York, NY, USA, 2008, pp. 257–266. http://doi.acm.org/10.1145/1367497.1367533.
[42] K. Church, B. Smyth, Understanding the intent behind mobile information needs, in: Proceedings of the 14th international Conference on Intelligent User Interfaces, IUI '09, ACM, New York, NY, USA, 2009, pp. 247–256. http://doi.acm.org/10.1145/1502650.1502686.
[43] M. Henzinger, B.-W. Chang, B. Milch, S. Brin, Query-free news search, World Wide Web 8 (2005) 101–126. http://dx.doi.org/10.1007/s11280-004-4870-6.
[44] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446. http://doi.acm.org/10.1145/582415.582418.